

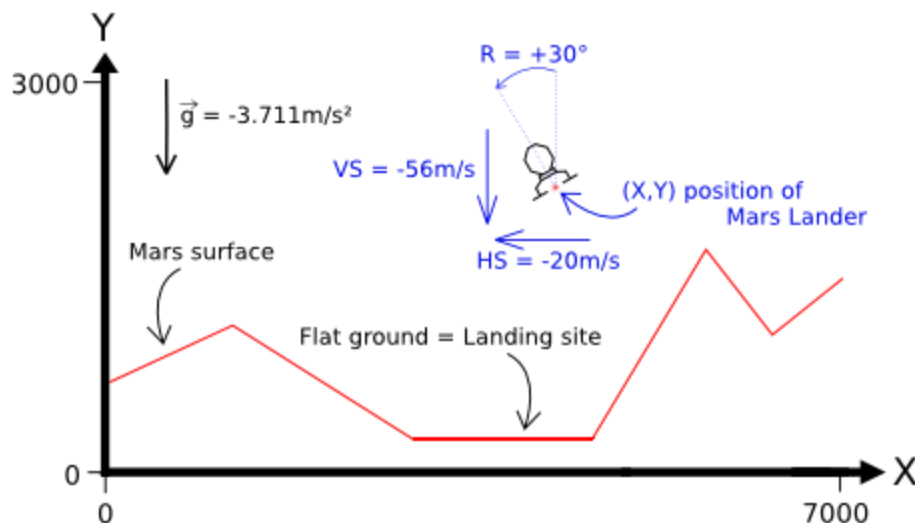
The Goal

The goal for your program is to safely land the "Mars Lander" shuttle, the landing ship which contains the Opportunity rover. Mars Lander is guided by a program, and right now the failure rate for landing on the NASA simulator is unacceptable.

Note that it may look like a difficult problem, but in reality the problem is easy to solve. This puzzle is the first level among three, therefore, we need to present you some controls you won't need in order to complete this first level.

Rules

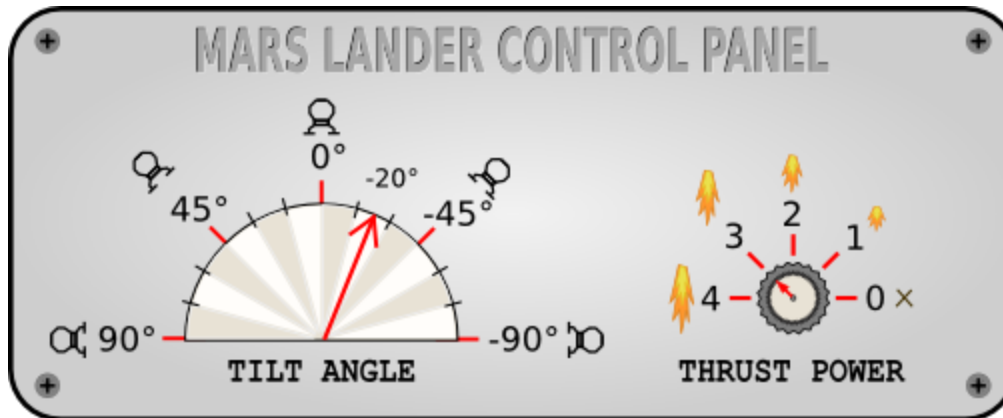
Built as a game, the simulator puts Mars Lander on a limited zone of Mars sky.



The zone is 7000m wide and 3000m high.

For this level, Mars Lander is above the landing zone, in vertical position, with no initial speed.

There is a unique area of flat ground on the surface of Mars, which is at least 1000 meters wide. Every second, depending on the current flight parameters (location, speed, fuel ...), the program must provide the new desired tilt angle and thrust power of Mars Lander:



Angle goes from -90° to 90° . Thrust power goes from 0 to 4 .

For this level, you only need to control the thrust power: the tilt angle should be 0. The game simulates a free fall without atmosphere. Gravity on Mars is 3.711 m/s^2 . For a thrust power of X, a push force equivalent to $X \text{ m/s}^2$ is generated and X liters of fuel are consumed. As such, a thrust power of 4 in an almost vertical position is needed to compensate for the gravity on Mars.

For a landing to be successful, the ship must:

- land on flat ground
- land in a vertical position (tilt angle = 0°)
- vertical speed must be limited ($\leq 40 \text{ m/s}$ in absolute value)
- horizontal speed must be limited ($\leq 20 \text{ m/s}$ in absolute value)

Remember that this puzzle was simplified:

- the landing zone is just below the shuttle. You can therefore ignore rotation and always output 0 as the target angle.
- you don't need to store the coordinates of the surface of Mars to succeed.
- you only need your vertical landing speed to be between 0 and 40 m/s – your horizontal speed being nil.
- As the shuttle falls, the vertical speed is negative. As the shuttle flies upward, the vertical speed is positive.

Note

For this first level, Mars Lander will go through a single test.

Tests and validators are only slightly different. A program that passes a given test will pass the corresponding validator without any problem.

Game Input

The program must first read the initialization data from standard input. Then, within an infinite loop, the program must read the data from the standard input related to Mars Lander's current state and provide to the standard output the instructions to move Mars Lander.

Initialization input

Line 1: the number surfaceN of points used to draw the surface of Mars.

Next surfaceN lines: a couple of integers landX landY providing the coordinates of a ground point. By linking all the points together in a sequential fashion, you form the surface of Mars which is composed of several segments. For the first point, landX = 0 and for the last point, landX = 6999

Input for one game turn

A single line with 7 integers: X Y hSpeed vSpeed fuel rotate power

- X,Y are the coordinates of Mars Lander (in meters).
- hSpeed and vSpeed are the horizontal and vertical speed of Mars Lander (in m/s). These can be negative depending on the direction of Mars Lander.
- fuel is the remaining quantity of fuel in liters. When there is no more fuel, the power of thrusters falls to zero.
- rotate is the angle of rotation of Mars Lander expressed in degrees.
- power is the thrust power of the landing ship.

Output for one game turn

A single line with 2 integers: rotate power :

- rotate is the desired rotation angle for Mars Lander. Please note that for each turn the actual value of the angle is limited to the value of the previous turn +/- 15°.
- power is the desired thrust power. 0 = off. 4 = maximum power. Please note that for each turn the value of the actual power is limited to the value of the previous turn +/- 1.

Constraints

$2 \leq \text{surfaceN} < 30$

$0 \leq X < 7000$

$0 \leq Y < 3000$

$-500 < \text{hSpeed}, \text{vSpeed} < 500$

$0 \leq \text{fuel} \leq 2000$

$-90 \leq \text{rotate} \leq 90$

$0 \leq \text{power} \leq 4$

Response time per turn $\leq 100\text{ms}$

Example

Initialization input

6 (surfaceN) Surface made of 6 points

0 1500 (landX landY)

1000 2000 (landX landY)

2000 500 (landX landY) Start of flat ground

3500 500 (landX landY) End of flat ground
5000 1500 (landX landY)
6999 1000 (landX landY)

No output expected

You can ignore this but you need to read the values.

Input for turn 1

2500 2500 0 0 500 0 0 (X Y hSpeed vSpeed fuel rotate power)

Output for turn 1

0 3

Input for turn 2

2500 2499 0 -3 499 0 1 (X Y hSpeed vSpeed fuel rotate power)

Output for turn 2

0 3

Input for turn 3

2500 2495 0 -4 497 0 2 (X Y hSpeed vSpeed fuel rotate power)

Output for turn 3

0 2

Mars Lander - LEVEL 1 - FIRST CONTACT

Cape Canaveral, June 21st 2003, 2 a.m. in the morning. Loud voices echo through the deserted corridors of the Kennedy Space Center.

“For god's sake Mike! Take off is in less than a month and we still don't have a reliable solution!”

Square jaws and a military haircut, Jeff is the manager of the Mars Opportunity mission and right now he's giving his chief engineer a hell of a time.

“ But simulations show a success rate of 99%. ”

“ So what? 1%, that's huge! Would you bet the success of the mission on that? And keep in mind, if Opportunity crashes, you and I, we'll both be done for. ”

“ The top NASA engineers have already worked on this project. It's the best we can achieve! ”

“ Then, let's try with external resources and see if anything better comes out of it. ”